Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning

Blad, Christian; Koch, Søren; Ganeswarathas, Sajuran; Kallesøe, Carsten; Bøgh, Simon

Published in:
29th International Conference on Flexible Automation and Intelligent Manufacturing

DOI (link to publication from Publisher):
10.1016/j.promfg.2020.01.159

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 38 (2019) 1308–1315

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

Control of HVAC-systems with Slow Thermodynamic Using Reinforcement Learning

C. Blad[b,d]*, S. Koch[a], S. Ganeswarathas[a], C.S. Kallesøe[c,d], S. Bøgh[a,b]

[a]*Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*
[b]*Robotics & Automation Group, Dept. of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg Øst, DK-9220, Denmark*
[c]*Dept. of Electronic systems, Aalborg Unicersity, Fredrik Bajersvej 7, Aalborg Øst, DK-9220, Denmark*
[d]*Grundfos A/S, Poul Due Jensens Vej 7, 8850 Bjerringbro*

Abstract

This paper proposes an adaptive controller based on Reinforcement Learning (RL), which copes with HVAC-systems consisting of slow thermodynamics. Two different RL algorithms with Q-Networks (QNs) are investigated. The HVAC-system is in this study an underfloor heating system. Underfloor heating is of great interest because it is very common in Scandinavia, but this research can be applied to a wide range of HVAC-systems, industrial processes and other control applications that are dominated by very slow dynamics. The environments consist of one, two, and four zones within a house in a simulation environment meaning that agents will be exposed to gradually more complex environments separated into test levels. The novelty of this paper is the incorporation of two different RL algorithms for industrial process control; a QN and a QN + Eligibility Trace (QN+ET). The reason for using eligibility trace is that an underfloor heating environment is dominated by slow dynamics and by using eligibility trace the agent can find correlations between the reward and actions taken in earlier iterations

Keywords: Sustainable Manufacturing Engineering and Resource-Efficient Production; Artificial Intelligence in Manufacturing; Modelling and Simulation; HVAC-Systems.

* Corresponding author. Tel.: +45 29320242
E-mail address: Cblad@m-tech.aau.dk

2351-9789
10.1016/j.promfg.2020.01.159

1. Introduction

To cope with rising energy demands and an ambition to reduce the carbon footprint from heat and energy production, regulation regarding insulation of buildings has increased. Another way to reduce energy consumption of buildings is to use more advanced controllers, which reduce energy waste and increase comfort. For large buildings, Model Predictive Controllers (MPCs) have showed to be effective [1], but an MPC requires a full thermodynamic model of the building, which for normal households is not economically feasible to make.

A traditional controller for an underfloor heating system in a household is a hysteresis control with the room temperature as input. This controller opens and closes for the control valve supplying heat to the floor dependent on the room temperature. The core issues with using hysteresis control with the room temperature as input for controlling the room temperature is the slow thermodynamic properties of the floor, which can result in time constants between 10 minutes to 3 hours depending on the floor type and material. Because of the delayed responses in the system a hysteresis controller is not able to keep the temperature constant because of its inability to predict the energy need for the room.

This paper suggests an adaptive controller based on reinforcement learning with a neural network. Reinforcement learning is, like animal learning, based on learning by interacting with a given environment [2]. Because its learning capabilities, reinforcement learning-based control naturally adapts, to whatever environment it interacts with. Furthermore, the reinforcement learning algorithms suggested in this paper are also model-free, which, as stated earlier, is necessary for the controller to be economically feasible. In this paper two algorithms are tested in four simulation environments. Two conclusions will be derived from this; 1) by adding eligibility trace the algorithm will perform better in an environment dominated by slow dynamic, 2) by increasing the complexity of the state-action space the algorithm will become unstable and therefore limit the use to smaller state-action spaces.

2. Use case

To identify the initial problem a sketch of an underfloor heating system is shown in Fig. 1, illustrating heat fluxes in a room. In Fig. 1 the temperature of the water running through the pipes in the floor is controlled by a mixing unit. This mixing unit can be a thermostatic mixing unit or an electromechanically actuated mixing unit. A thermostatic mixing unit can be outdoor compensated, meaning it adjusts the temperature of the mixing water according to the outside temperature – high outside temperature, low mixing temperature and vice versa. An electromechanically actuated mixing unit is less common because it needs a control input, but it does allow for more control of the environment. In the work presented in this paper an electromechanical valve will be used due to its flexibility. The electromechanical valve is controlled by a step size controller. The control agent will still control the mixing temperature, but the incremental change in the temperature will be adjusted according to the distance to the given reference temperature. Meaning, if the distance between the room and reference temperature distance is high, the incremental change will be high and vice versa.

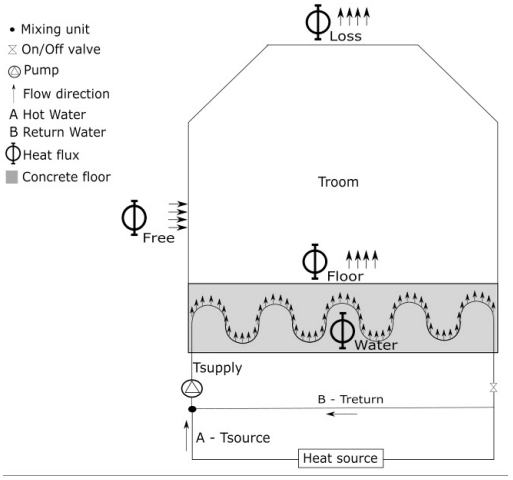

Fig. 1. An underfloor heating system with one temperature zone consisting of four heat fluxes Φ. Heat fluxes are in the simulation calculated with a 1D heat differential equation, the free heat flux Φ_{Free} is set to zero in all simulations. The hydraulic system of the underfloor floor heating is presented in the bottom of the figure, and it is assumed there is a local heat source, which also could have been a district heating source.

By using a reinforcement learning based controller it is theoretically possible for an agent to adapt to the thermodynamic properties of a given thermal zone, by making an internal predictive model and using predictive external data such as weather forecasts to enhance performance. In this paper, it is demonstrated that it is possible for the agent to learn to control a four-zoned underfloor heating system in a simulation environment in Simulink within a tolerance of 1 °C. The simulation environments have the thermal dynamic properties of an underfloor heating system, but is only affected by the ambient temperature, so no sun, wind etc. and there is no thermal transfer between interior walls.

3. Reinforcement Learning

This paper will use aspects of the deep reinforcement learning algorithm Deep Q-network (DQN), which was developed as a method to combine deep neural networks and reinforcement learning for learning directly from high-dimensional sensory inputs [3]. An underfloor heating system can in contrast to Atari games, which the DQN was developed for, be described as systems with low-dimensional sensory inputs. Therefore, the proposed algorithm utilizes a neural network with one hidden layer. The neural network with weights θ is used as a function approximator to approximate the action value function $Q(s, a) \approx Q(s, a, \theta)$. The reason for using a function approximator is because it is not computational efficient to store a large Q-table. As one might suspect using function approximators does also come with drawbacks [4]. Especially using nonlinear function approximators such as a neural network has proven hazards because of the risk of instability or divergence [5]. Using experience replay and fixed target Q in the DQN has proven that a neural network can be an efficient and stable function with improved convergence behavior [3].

Since the DQN was developed improvements has been made to the experience replay method, these include prioritized experience replay[6] and hindsight experience replay[7]. These techniques have not been considered for this paper.

Experience replay works by storing the experience from a given iteration at time t with the stats s, action a, reward r and the next state s_{t+1}, $e_t = (s_t, a_t, r_t, s_{t+1})$. The experience is stored in a memory $D[e_1, . , . , e_t]$ and used to update the weights in the Q-network through a loss function and an optimizer. The purpose of experience replay is to reduce correlation between observation by randomly drawing experience from the matrix D, this also enables the agent to use rare experience more than ones[8]and thereby learn more efficiently from a limited amount of data. The loss function used to calculate the error between target Q and predicted Q is derived from the bellman equation and can be expressed by the following equation:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(Q_{target} - Q_{predicted}\right)^2\right] \quad (1)$$

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right) - Q(s, a; \theta_i)\right)^2\right] \quad (2)$$

An optimizer is used to update the weights in the neural network. In the original DQN is stochastic gradient descent used, in this algorithm is the Adam optimizer is used. The Adam optimizer is a first order gradient-based optimizer like stochastic gradient descent, but it also uses estimations of lower-order moments, which makes it suitable for non-stationary objectives and noisy and/or sparse gradients like a neural network can have [9].

To ensure that the agent during training explores the state action space and exploits what it has already learned, a Softmax function is used as an action selector. The Softmax function works by setting the parameter τ which indicates the agent's level of confident. τ =0 indicates full confident and to encourage more exploration τ is increased. The mathematical description of the Softmax function can be seen in the following equation: $\left(Q(s_t, a_t)\right)^{\tau}$

$$W_s: a \in A \rightarrow \frac{e^{\left(\frac{Q(s_t,a_t)}{\tau}\right)}}{\sum a_{t+1} e^{\left(\frac{Q(s_t,a_{t+1})}{\tau}\right)}} \; with\; \tau > 0 \quad (3)$$

As stated in the abstract it is of interest to test if an eligibility trace implementation can perform well in an environment dominated by slow dynamics such as underfloor heating systems. Eligibility trace is a method that makes it possible to make a trade-off between Monte Carlo and Temporal Difference, where Monte Carlo has high variance because Monte Carlo only updates at the end of the episode. This MDP is considered continues as it does not have episodes, so Monte Carlo cannot be used. Temporal Difference on the other hand updates for every iteration but uses its own estimation to update, which means it has bias [2]. Eligibility trace or n-step learning uses a parameter n, where n is the number of iterations that will pass before an update is made. This means if n is the same size as the number of iterations in an episode it is Monte Carlo.

For experience replay to be compatible with eligibility trace, a few modifications has been made to way the data is drawn from the experience memory D. Instead of drawing random samples, the agent is drawing random batches of the same size as the eligibility trace

4. Experiment

The structure of the used Q-network with input states, Q-values, and the action selector, is illustrated in Fig. 2.

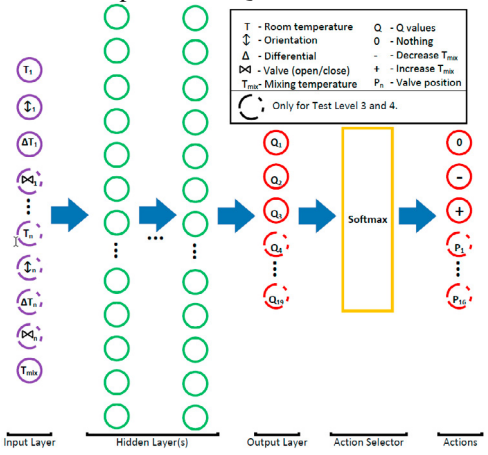

Fig. 2. Illustration of the neural network, with input states, Room temperature, orientation to reference temperature, differential of room temperature, valve position, and mixing temperature. And output Q values and the Softmax action selector. Two hidden layers is shown in the illustration, this is just to illustrate that it is possible to add more layers.

The experiments have been divided into four different test levels (TL's) to sort out which RL algorithm that is suited for the task. Both algorithms will start in the environment of TL1 and then continue to TL4 increasing the complexity where they need to satisfy the requirement (R): Room temperature(s) is allowed a standard deviation of 1 °C from the room reference temperature 22 °C. The TL1 environment consist of one temperature zone, but with no thermo-properties of an underfloor heating system. The only task of the agent is to control the mixing temperature. The environment of the TL2 is still one temperature zone but with the thermo-properties of an underfloor heating system has been added in this test level. By doing this it will be possible to investigate the effect of eligibility trace in a dynamic environment. TL3 and TL4 consist of multiple zones meaning the agent can control valves and the mixing temperature. TL3 has two zones and TL4 has four zones. The ambient temperature in the simulation environment is set to be constant in the beginning of the simulation and then a 1-day and a 14-days sine cycle is added to represent day and night changes and longer changes over 14 days. Hyperparameters and settings of the algorithms are shown in Appendix A. The following results from the test levels will consist of response temperature plot of the environment from 232 days' time per test in TL1 and TL2 and 926 days' time in TL3 and TL4.

4.1. Test Level 1 Results

Two tests were performed in TL1, one test of the QN and one test of the QN+ET, the results can be seen in Fig. 3.

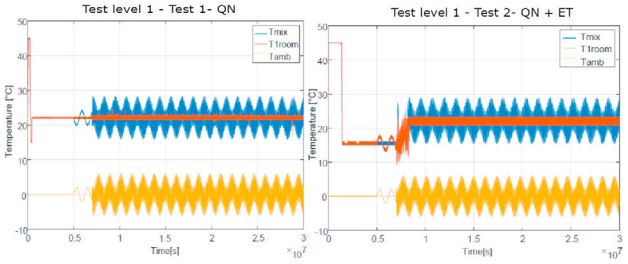

Fig. 3. Results of Test Level 1 with no thermo-properties of an underfloor heating system has been introduced. The mixing temperature T_{mix} is blue, room temperature $T1_{room}$ is red and ambient temperature T_{amb} is yellow.

It can be seen from the results of the two tests in Fig. 3 that the QN algorithm without eligibility trace does perform equally good or better than the algorithm with eligibility trace. The reason for this is that there are no slow responses in this system, because there is no reason that eligibility trace should improve performance.

4.2. Test Level 2 Results

Two tests are performed in TL2, where Test 1 is the QN algorithm and test 2 is the QN algorithm with eligibility trace, the results of the two tests are shown in Fig. 4.

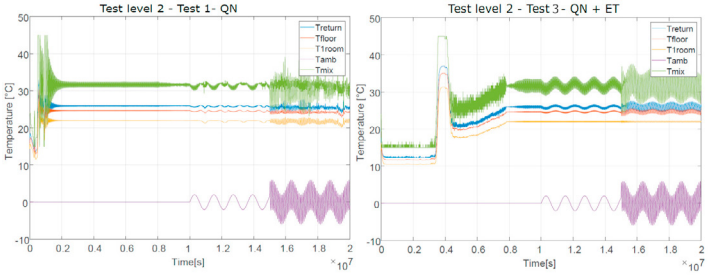

Fig. 4. Results of Test Level 2 with thermo-properties of an underfloor heating system has been introduced. The mixing temperature T_{mix} is green, ambient temperature T_{amb} is purple, return temperature T_{return} is blue, floor temperature T_{floor} is red and room temperature $T1_{room}$ is yellow.

Table 1. Satisfaction of requirement R from the period $1.8 \cdot 10^7$ to $2 \cdot 10^7$ seconds in Test Level 2 are grey scaled elements with tests where R is satisfied.

Test	Model type	$[\bar{x}, \sigma_x]$
1	QN	[21.52,0.54]
2	QN	[22.44,0.54]
3	QN+ET	[22.04,0.11]
4	QN+ET	[22.09,0.47]

From the results in Table 1 and Fig. 4 it can be seen QN+ET performs best, and that the QN algorithm does not meet the requirements in the first test but manages to do so in second test. A comparison to TL1 without slow dynamics reveals that the QN algorithm performed slightly better than QN+ET. This comparison leads to the conclusion that ET does improved performance when the system is dominated by slow dynamics.

4.3. Test Level 3 Results

TL3 consists of four tests; two with the QN algorithm and two with the QN+ET algorithm. The results of two successful tests are shown in Fig. 5.

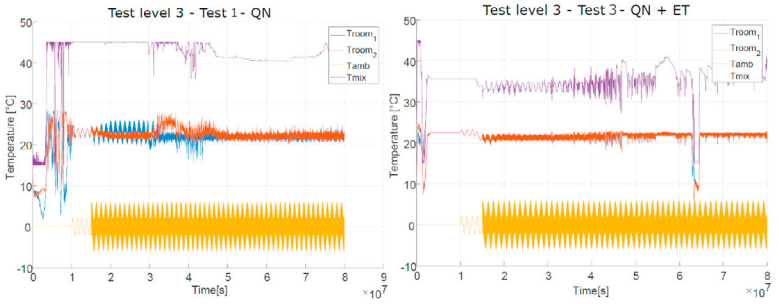

Fig. 5. Results of Test Level 3 with thermo-properties of an underfloor heating system has been introduced. On the left Test 1 and Test 3. Where T_{room} 1 is blue, T_{room} 2 is red, T_{mix} is purple and the ambient temperature T_{amb} is yellow

From Table 4 it is seen that the QN+ET algorithm satisfies the requirement two times in both rooms, where the QN algorithm only satisfied the requirements for one of the two tests and it did not perform as well in this test regarding standard deviation or mean value. Note that Test 2 with QN+ET also manages to have the lowest energy usage due to average lowest mixing temperature T_{mix}.

Table 2. Satisfaction of requirement R from the period $7 \cdot 10^7$ to $8 \cdot 10^7$ seconds in Test Level 3 with QN and QN+ET where grey scaled elements are tests where R is satisfied.

Test	Model type	$T1_{room}[\bar{x}, \sigma_x]$	$T2_{room}[\bar{x}, \sigma_x]$	$T_{mix}[\bar{x}]$
1	QN	[20.12,2.11]	[21.81,1.67]	[40.47]
2	QN	[22.38,0.62]	[22.35,0.63]	[38.74]
3	QN+ET	[21.91,0.34]	[21.87,036]	[36.61]
4	QN+ET	[22.03,0.58]	[22.19,0.55]	[42.73]

4.4. Test Level 4 Results

TL4 consists of three tests with the QN+ET algorithm due to it satisfied the requirement in TL3 and it was not possible to perform a satisfied test of the QN algorithm. The result of TL4 is shown in Fig. 6.

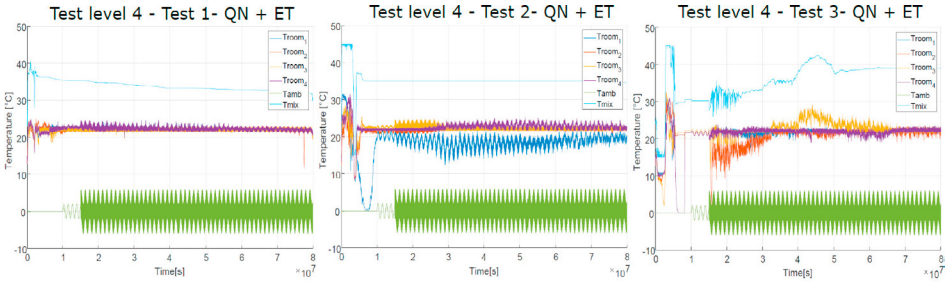

Fig. 6. Results from three tests from Test Level 3 of QN+ET.

From Table 4 it is observed that Test 3 is successful, it did however require three tests, which shows that the algorithm has become unstable due to the complexity of the state-action space.

Table 3. Satisfaction of requirement R from the period $7{\cdot}10^7$ to $8{\cdot}10^7$ seconds in Test Level 4 with QN+ET where grey scaled elements are tests where R is satisfied.

Test	Model type	$T1_{room}$ $[\bar{x}, \sigma_x]$	$T2_{room}$ $[\bar{x}, \sigma_x]$	$T3_{room}$ $[\bar{x}, \sigma_x]$	$T4_{room}$ $[\bar{x}, \sigma_x]$	T_{mix} $[\bar{x}]$
1	QN+ET	[21.89,0.36]	[21.77,1.13]	[21.99,0.37]	[21.99,0.38]	[32.13]
2	QN+ET	[19.69,0.85]	[22.07,0.27]	[22.2,0.25]	[22.63,0.6]	[34.73]
3	QN+ET	[22.08,0.26]	[21.95,0.61]	[22.2,0.28]	[22.35,0.34]	[39.05]

5. Conclusion

By reviewing the results of the 9 tests it can be concluded that it is possible for a reinforcement learning based controller to control the designed simulation environment of an underfloor heating system. In this study two different algorithms have been tested; 1) QN and 2) QN +ET. By comparing the performance of the two algorithms it can be concluded that the eligibility trace addition to the Q-network does increase performance slightly, but only when there are slow dynamics included in the simulation. Furthermore, it is concluded that the function approximator does become unstable when increasing the complexity of the state-action space. This means that one should be aware of the size of the state-action space when using this technique and additional research is needed to make the proposed reinforcement learning approach more robust.

6. Future Work

As stated in the introduction, the simulation environment is simplified i.e. it does not include interior wall transfer, windows, sun, wind etc. A more detailed simulation environment would make it possible to take all these parameters into account. To utilize the full potential of reinforcement learning it would be ideal to use forecasted weather data. This way the agent would not only depend on its internal model of the dynamic behavior, especially in an environment dominated by long delayed response this would be desirable. In this study the simulation environment has been defined as one single MDP, which can include 1, 2 and 4 temperature zones. It has been observed that by increasing the number of zones in the MDP the training time becomes longer and the agent does perform less desirable. This, of cause, makes sense, because the complexity of the state-action space is increased with the number of zones. It would be interesting to explore the possibility to design an agent with multiple MDPs,

or a multi-agent controller, to control the temperature zones independently. All the above observations are subjects for future works.

The proposed reinforcement learning control is not robust enough to use in commercial applications yet. Additional research must be made into increasing the robustness of the controller. Multiple improvements have been made to the DQN algorithm, the latest is the Rainbow algorithm [10]. These improvements might also be better suited for the current and future tasks at hand.

References

[1] SamuelPŕívara and Jan Širokýb and Lukáš Ferkla and Jiří Ciglera *Model predictive control of a building heating system: The first experience*. Energy and Buildings Volume 43, Issues 2–3, Pages 564-572, February–March 2011

[2] Richard S. Sutton and Andrew G. Barto *Reinforcement Learning: An Introduction*. 2. Edition. 2018.

[3] Volodymyr Mnih and Koray Kavukcuoglu and David Silver and Andrei A. Rusu and Joel Veness and Marc G. Bellemare and Alex Graves and Martin Riedmiller and Andreas K. Fidjeland and Georg Ostrovskivand Stig *Human-level control through deep reinforcement learning*. Nature , 529–533.

[4] Sebastian Thrun and Anton Schwartz *Issues in Using Function Approximation for Reinforcement Learning*. Proceedings of the Fourth Connectionist Models Summer School Lawrence Erlbaum Publisher, Hillsdale, NJ, Dec. 1993

[5] John N. Tsitsiklis and Benjamin Van Roy. *An Analysis of Temporal-Difference Learning. TRANSACTIONS ON AUTOMATIC CONTROL*. Transactions on automatuc control VOL. 42, NO. 5, May 1997.

[6] Tom Schaul, John Quan, Ioannis Antonoglou, David Silver. *Prioritized Experience Replay*. International Conference for Learning Representations, 2016.

[7] Andrychowicz, Marcin and Wolski, Filip and Ray, Alex and Schneider, Jonas and Fong, Rachel and Welinder, Pe ter and McGrew, Bob and Tobin, Josh and Pieter Abbeel, OpenAI and Zaremba, Wojciech *Hindsight Experience Replay*. Advances in Neural Information Processing Systems 30, pp. 5048–5058, 2017.

[8] Long-Ji Lin. *Self-improving reactive agents based on reinforcement learning, planning and teaching*. Machine Learning, Volume 8, Issue 3–4, pp 293–321 May 1992,

[9] Diederik P. Kingma, Jimmy Lei Ba. *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. International Conference for Learning Representations, 2015.

[10] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, David Silver *Rainbow: Combining Improvements in Deep Reinforcement Learning*. The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).

Appendix A. Settings of Algorithms for Test Levels

Table A1. Standardization of input variables.

Input variables	Standardisation Equations
T_{std}	$\frac{T}{35}$
T_{std}	$\lvert (T\text{-}T_{last})\cdot 10) \rvert$
T_{std}	$\frac{0.5 \; if \; T_{room} \geq T_{ref}}{0.5 \; if \; T_{room} < T_{ref}}$
T_{std}	$\frac{1 \; if \; Q > 0}{0 \; if \; Q = 0}$
T_{std}	T_{std}

Table A2. Setup for Q-networks for test levels.

Q-network Setup	Test Level			
	1	2	3	4
Input variables	4	4	9	17
Hidden layers	1	1	1	1
Hidden neurons per layer	30	30	30	30
Output variables	3	3	7	19

Table A3. Hyperparameters for algorithms in test levels.

Hyperparameters	Algorithm	
	QN	QN+ET
Learning rate T_{std}	0.001	0.001
Discount factor γ	0.9	0.9
Softmax temperature τ	100	100
Experience replay batch size	50	50
Experience replay capacity	100000	100000
Eligibility trace steps n	-	30